LaTeX de A a B

uma introdução não tão completa.

Gabriel Dias Pais
Leandro Augusto Ferreira

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

O projeto: *O principal objetivo deste material é apresentar informações e exemplos úteis sobre tipografia em LaTeX para auxiliar usuários iniciantes e intermediários. Esse material foi criado para um mini curso ministrado no XI SIM 2008 - XI Simpósio da Matemática para a Graduação no ICMC-USP.*

São Carlos - Agosto/2008

Sumário

Capítulo 1

Introdução

A primeira pergunta que alguém se fará, ao se deparar com o LaTeX, é sobre a vantagem de se usá-lo, ao invés de ferramentas *WYSIWYG* [1] como o *OpenOffice Writer* e o *Microsoft Word*, e muitas pessoas comparam o LaTeX com esses programas. Na verdade, o LaTeX e as ferramentas *WYSIWYG* possuem objetivos e base de usuários completamente distintos, não sendo prudente afirmar qual modelo seja superior. São apenas diferentes. É necessário, entretanto, mostrar as vantagens e desvantagens de se utilizar um e outro para que as pessoas saibam qual modelo as convém.

Vantagens

- A principal vantagem, e de onde as outras tiram sua validade, é o algoritmo avançado do TeX, que permite criar documentos de aparência verdadeiramente profissional.
- O LaTeX encoraja as pessoas a concentrar suas atenções no conteúdo e na distribuição lógica das idéias, e não na aparência, resultando em textos bem estruturados.
- É possível utilizar o LaTeX sem efetivamente aprender muitos comandos, especialmente com a utilização conjunta de ferramentas WYSIWYG como o LyX.
- A edição de fórmulas matemáticas é robusta e sua apresentação, visualmente agradável.
- Estruturas tipográficas complexas como bibliografia, tabela de conteúdo e citações estão abstraidas, podendo ser geradas facilmente e de forma consistente ao longo do documento.
- Ambos TeX e LaTeX são programas livres, permitindo que existam versões para praticamente todo sistema operacional disponível.
- Utilização modesta dos recursos do sistema. É possível editar e imprimir mesmo em um computador remoto, através de uma sessão SSH ou telnet.

[1] Do inglês: "What You See Is What You Get" que significa "O que você vê é o que você tem"

O processador também não precisa trabalhar todo o tempo para criar a formatação do documento, pois a diagramação só é feita uma vez, após o término da edição.

Desvantagens

- A principal desvantagem é que, embora a utilização de estilos prontos de documento seja fácil, a criação de novos modelos leva muito tempo, sendo que nem sempre é possível encontrar modelos de documento que estejam em conformidade com o requerido por muitas instituições.
- A aprendizagem é mais difícil que em programas WYSIWYG, pois embora a estrutura lógica do documento seja intuitiva, os comandos do LaTeX, obviamente, não o são.

Esclarecidas essas primeiras dúvidas, vamos descobrir o que significa afinal TEX e LATEX.

1.1 O que significa TEX ?

TEX (Tau épsilon chi pronunciada como "latequi") é um sistema de tipografia criado por Donald Knuth. É popular no meio acadêmico, principalmente entre os físicos, matemáticos e cientistas da computação, devido a sua capacidade de produzir fórmulas e símbolos matemáticos de uma forma elegante.

1.2 O que significa LATEX ?

LATEX é um conjunto de macros para o processador de textos TEX, utilizado amplamente para a produção de textos matemáticos e científicos por causa de sua alta qualidade tipográfica. Entretanto, também é utilizado para produção de cartas pessoais, artigos e livros sobre assuntos muito diversos.

Como um conjunto de macros para o TEX, o sistema LATEX fornece ao usuário um conjunto de comandos de alto nível, sendo, dessa forma, mais fácil a sua utilização por pessoas nos primeiros estágios de utilização desse sistema. Possui abstrações para lidar com bibliografias, citações, formatos de páginas, referência cruzada e tudo mais que não seja relacionado ao conteúdo do documento em si.

O LATEXfoi desenvolvido na década de 80, por *Leslie Lamport*, estando, atualmente, na versão denominada (LATEX2).

1.3 Um pouco de história

Uma pequena biografia dos matemáticos e criadores das ferramentas que revolucionaram a tipografia e humilharam as descobertas de Gutemberg [2].

[2] Johannes Gutemberg foi um inventor alemão que se tornou famoso pela sua contribuição para a tecnologia da impressão e tipografia. Inventou uma liga para os tipos de metal, tintas à base de óleo, além de uma prensa gráfica inspirada nas prensas utilizadas para espremer as uvas no fabrico do vinho.

Donald Knuth - Donald Ervin Knuth (Milwaukee, 10 de Janeiro de 1938) é um cientista computacional de renome e professor emérito da Universidade de Stanford. Formado em matemática é o autor do livro *The Art of Computer Programming*, uma das principais referências da ciência da computação. Ele praticamente criou o campo análise de algoritmos e fez muitas das principais contribuições a vários ramos da teoria da computação. Ele também criou o sistema de tipográfico TEX, o sistema de criação de fontes *METAFONT*, além de ser pioneiro do conceito de programação literária.

Leslie Lamport - Leslie Lamport (nascido em 1941) é um cientista computacional norte-americano Lamport formou-se em Matemática pelo Massachusetts Institute of Technology em 1960, com mestrado e doutorado em Matemática pela Brandeis University, concluídos respectivamente em 1963 e 1972. O tema de sua tese de doutorado foram singularidades em equações diferenciais parciais analíticas. Após a graduação, ele começou sua carreira como cientista computacional no Massachusetts Computer Associates, SRI International, Digital, e Compaq. Em 2001, ele se juntou à Microsoft Research em Mountain View, na California. Suas pesquisas contribuíram com a fundação da teoria de sistemas distribuídos. Fora da ciência computacional, o Dr. Lamport é melhor conhecido como sendo o desenvolvedor inicial do sistema de preparação de documentos, o LATEX.

Como tudo começou ... No ano de 1977, Donald Knuth estava discutindo e experimentando a editoração de um livro pelo computador. Neste mesmo ano, a editora encarregada do segundo volume do seu livro *The Art of Computer Programming*, enviou um *preprint* do livro. Decepcionado pela sua baixa qualidade, Kunuth decidiu concretizar sua idéia: Implementar um sistema computacional eficiente para editoração, hoje conhecido como sistema "TeX". Como não existia uma fonte adequada para documentos técnicos e científicos, ele decidiu fazer design das fontes e implementar um sistema de fontes eficiente para ser usado no TEX. Esta fonte é conhecido como fonte "Computer Modern" e o sistema de fontes é conhecido como *METAFONT*. O sistema TEX ficou pronto em 1982 e o sistema de metafont em 1984. A fonte Computer Modern foi concluída um ano depois, em 1985. O macro (configuração) que *Kunuth* usou para editar o seu livro foi distribuído junto com TEX e recebeu o nome de "plain TEX". O sistema TEX é formado pelo conjunto de programas para processar o documento de forma automática e eficiente, mas precisava do conhecimento de editoração para produzir documentos de qualidade. Com o intuito de permitir diagramação dos documentos científicos de qualidade profissional sem que necessite do conhecimento específico da editoração, foi iniciado um projeto para desenvolver macros (configurações) novos para TEX. O grupo foi coordenado pelo matemático *Leslie Lamport* do DEC (Digital Equipment Corporation), especialista em computação (LTA - lógica temporal de ações, usado na modelagem dos para sistemas concorrentes e reativos). O projeto foi concluído em 1985 e recebeu o nome de "LaTeX". O projeto LATEX2 liderado por *Frank Mittelbach* foi responsável pela melhoria do LATEX desenvolvido por *Lamport*, corrigindo e incrementando a macro. O grupo de LATEX3 lança-o em 1994.

1.3.1 Os fatos na ordem cronológica

1977: Motivado pela má qualidade editorial do seu livro "A Arte de Programação do Computador", volume 2, Kunuth resolve implementar o sistema TEX.

1978: Primeiro protótipo do TEX.

1978: Iniciou o projeto de MetaFont (formato de fonte do TEX).

1979: Iniciou o projeto Euler para desenvolver fontes matemáticos (Kunuth fazia parte deste grupo).

1980: Primeiro protótipo da fonte "Computer Modern".

1980: Foi fundado o TUG (grupo de usuário de TEX).

1981: Primeira versão do sistema Web (sistema de processamento de documentos usado pelo TEX). Após alguma melhoria, receberá a denominação de Web83.

1982: Primeira versão do TEX (versão 0).

1982: Kunuth começa a escrever o livro "The TeX Book".

1983: TEX versão 1.

1984: "The TeX Book" (Donald Kunuth) foi publicado.

1984: Primeira versão do MetaFont (versão 0).

1984: Kunuth começa a escrever o livro "MetaFont Book".

1985: A fonte "Computer Modern" concluída.

1985: Lamport finaliza o LATEX (versão 2.09).

1986: TEX versão 2.

1986: "The Metafont Book" (Donald Kunuth) foi publicado.

1989: Agora TEX suporta caracteres de 8 bits.

1990: TEX versão 3.

1994: LATEX2 e foi lançado pelo grupo de "LaTeX3" (coordenado pelo Frank Mittelbach).

1994: O livro "A Document Preparation System" (Laslie Lamport) sobre LATEX 2.09 foi publicado.

1994: O livro "The LaTeX Companion" (M. Goosses, F. Mittelbach e A. Samarin) sobre LaTeX2e foi publicado.

2000: "Young Ryu" lança o conjunto de fontes Post Script com qualidade profissional baseado em Times, denominado de txfonts. Agora poderá usar a fonte Times para produzir documentos com qualidade tipográfica comparavel ao do Computer Modern, sem necessidade de recorrer às fontes comerciais.

1.3.2 Tendências atuais

Atualmente, o LaTeX conta com recursos para gerar o arquivo do formato PDF tais como DVIPDFM, PDFTeX e PDFLaTeX e existem diversos convertores para gerar arquivos de outros formatos tais como HTML e RTF (Rich Text Format aceito em MS Word e compatíveis). Além disso, também existem conversores para outros formatos LaTeX. Veja o site http://www.tug.org/utilities/texconv/ para detalhes. Existem outras macros tal como ConTeXt (macro de propósito geral tal como plain TeX) e classes de LaTeXtais como KOMA-script que é distribuído com diversos sistema TeX. Também existe o desenvolvimento de outros sistemas de diagramação de documentos profissionais tais como NTS (New Typeset system) e Jade (para processar documentos escritos em DSSSL (Document Style Semantics and Specification Language) do SGML/XML tal como DocBook (SGML é abreviatura do Standard Generalized Markup Language. XML é subconjunto de SGML. HTML também segue o padrão SGML/XML). Em termos de facilitar a digitação do documento LaTeX, existem diversos editores específicos que ajudam na elaboração de documentos, oferecendo integração com os acessórios, localização de erros, inserção de símbolos ou comandos, integração com visualizador de DVI, etc. Quem não tem conhecimento do LaTeX ou que não goste de ficar editando o código fonte, poderá contar com o editor do tipo *WYSIWYG* que processa o documento usando LaTeX. Exemplo destes editores são LyX (free - Linux e Win) e Scientific Word/Work Place (comercial - Windows). Note que o TeXEmacs (free - linux) e alguns outros editores científicos também poderão exportar código em LaTeX. Usando editores WYSIWYG como estes, poderá elaborar os documentos como no editor do Office e gerar o documento de alta qualidade tipográfica divido ao uso do LaTeX.

Assim, "não sei usar o LaTeX" já não é desculpa para deixar de produzir documentos com alta qualidade tipográfica. Observe que o conhecimento do LaTeX nestes editores WYSIWYG melhora o desempenho devido ao fato de eles aceitarem a seqüência de digitação dos comandos de LaTeX e também por permitir inserção de comandos do LaTeX diretamente no documento.

1.4 Instalando o LaTeX

A seguir um resumo básico de instalação.

1.4.1 Distribuições

Os documentos escritos para o LaTeX estão em texto simples, sem formatação alguma. Nesse sentido, é possível escrever documentos para o LaTeX em qualquer editor de texto, mesmo nos mais simples graficamente, como o *VI* ou o Bloco de notas. Não obstante essa facilidade de edição de arquivos LaTeX, recomenda-se a utilização de programas específicos, muitos deles sendo software livre, como o *Kile*. Além disso, o LaTeX funciona em diversas plataformas, existindo distribuições para muitos sistemas operacionais, a exemplo de MiKTeX, para Windows; *MacTeX*, para Mac OS X; e TeX Live (multi plataforma, incluindo Linux).

1.4.2 Instruções

Independente da escolha do editor é necessário instalar algumas ferramentas. De maneira simplificada, o editor converterá o seu "código" LATEX em um arquivo compactado com o *layout* de texto editado.

As instruções a seguir são para instalar o ambiente LATEX para Windows e devem ser feitas antes de baixar o estilo. Baixar e instalar os pacotes abaixo, na ordem citada:

a) **Ghostscript:** Necessário para operações em arquivos tipo PostScript. Execute a instalação deste software seguindo as opções padrões. Baixar o programa a partir de:

   `ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/AFPL/gs814/gs814w32.exe`

   e execute sua instalação. Seguir as opções padrão do software.

b) **GSView:** Permite a visualização de arquivos no formato PostScript PDF no Windows. Baixar a partir de:

   `ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/ghostgum/gsv47w32.exe.`

   Na janela Select Language, escolher English. Nas outras janelas, seguir a seleção do software, sem alterar as opções padrões.

c) **MiKTeX:** O MiKTEX É a alma do ambiente LATEX, responsável por compilar os arquivos de entrada em **.tex* do LATEX e gerar as saídas no formato DVI, PS ou PDF. Instale a versão 2.6 baixando do endereço:

   `http://sourceforge.net/project/platformdownload.php?group_id=10783`

   Estando o MikTEX instalado, aproveitar também para configurar o idioma português. No menu Iniciar do Windows na pasta MiKTEX 2.6 selecionar Settings. Ao abrir a janela, selecionar a aba Languages e marcar a língua portuguesa. Marcar também outros idiomas, se necessários. A hifenização será automaticamente ativada.

d) **Editor Gráfico:** Agora você pode instalar o Editor LATEX de sua preferência. Sugestões: *WinEdit*, *TeXnicCenter*. A instalação desses editores é usualmente simples, quase um "Next, next to finish". Siga as instruções do instalador.

   TeXnicCenter `http://www.toolscenter.org/`

   WinEdit `http://www.winedt.com/`

Tudo instalado, execute o editor que você escolheu e siga as instruções para configuração de pacotes, conforme as opções padrões de configuração.

1.5 A arte de saber encontrar ajuda

Todo profissional deve desenvolver algumas habilidades básicas para bem encaminhar-se. Uma delas, que considero importante, é saber como e onde procurar ajuda. Ninguém é capaz de memorizar tudo que aprende. Sempre chega o momento em que é necessário se lembrar daquele comando ou então descobrir

como criar macros próprias para facilitar o texto especificamente que você está escrevendo.

É grande a chance de que sua dúvida, independente do que seja, já tenha sido questionada e respondida por outras pessoas. O maior problema, dado o volume de informações publicadas na internet, é encontrar esta resposta rapidamente. **Don't Panic!** Abaixo você encontrará alguns links com informações úteis a qualquer hora.

Blogs

- O que eu vou aprendendo em LaTeX, [5] `http://aprendolatex.wordpress.com/`
- TeX user groups around the world, [4] `http://tug.org/usergroups.html`

Apostilas

- CTAN - Comprehensive Tex Archive Network, [3] *The Not So Short Introduction to LaTeX2* uma das apostilas mais difundidas pelo mundo

  `http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`
- Universiade Federal de Minas Gerais, material do professor *Reginaldo de Jesus Santos* [6]

  `www.mat.ufmg.br/~regi/topicos/intlat.html`
- Universidade da Federal da Paraíba, material do professor *Lenimar Nunes de Andrade* [1]

  `http://lief.if.ufrgs.br/pub/latex/breve.pdf`
- Universidade Estadual de Londrina, material do professor *Silvano Cesar da Costa* [2]

  `http://www2.uel.br/pessoal/silvano/Apostila.pdf`

Capítulo 2

Criando um texto

Depois de um grande capítulo de lero-lero. Agora chegou a hora de colocar as mãos na massa redigindo seu primeiro texto em LaTeX.

2.1 Estrutura básica de um texto.

Um texto escrito em LaTeX possui uma estrutura básica, que é formada por um preâmbulo e um corpo. O preâmbulo possui vários comandos, mas obrigatoriamente devemos utilizar o comando `\documentclass`, cuja a sintaxe é

```
\documentclass[opções]{classe}
```

onde *opções* é um parâmetro opcional, que pode conter informações sobre o tamanho e tipo de fontes, tipo de papel, etc, e o parâmetro *classe* é **obrigatório** e define o estilo do texto digitado (artigo, tese, livro, etc). Logo abaixo deste comando, podemos utilizar comandos globais, que atuam em todo o texto.

O corpo começa sempre com o comando `\begin{document}` e termina com o comando `\end{document}`. Entre esses dois comandos, fica o texto propriamente dito, junto a comandos locais, que possuem apenas valor local. Segue abaixo um esquema da estrutura de um texto escrito LaTeX.

```
\documentclass[opções]{classe}

Comandos globais

\begin{document}

Comandos locais

\end{document}
```

2.1.1 Tipos de texto

Como já dissemos o comando `\documentclass` possui o parâmetro *classe* que é obrigatório, e que define o estilo do texto. Os estilos mais comuns são *article* (artigo), *report* (tese), *letter* (carta), *book* (livro) e *slides* (apresentações).

2.1.2 Usando Pacotes

Para o uso de alguns recursos no LaTeX é necessário a utilização de pacotes. Por exemplo, na inserção de **imagens**, no uso de **tabelas** e na utilização de **índices** precisamos incluir os pacotes no preâmbulo do texto.

Use o comando `\usepackage{pacote}` no preâmbulo, de preferência logo abaixo do comando `\documentclass[opções]{classe}`. Segue abaixo os principais pacotes.

Inserção de imagens

```
\usepackage{graphicx}
\usepackage{wrapfig}
```

Tabelas

```
\usepackage{bigstrut}
```

Verbatim

```
\usepackage{Verbatim}
```

Bibliografia

```
\usepackage{setspace}
```

Endentação

```
\usepackage{indentfirst}
```

Índice remissivo

```
\usepackage{makeidx}
```

2.2 Formatação do texto

2.2.1 Tipos de letras

Vários tipos de letras são utilizadas em LaTeX, podemos alterá-los com os comandos que aparecem nas duas primeiras colunas da tabela abaixo, que produzem os tipos listados na última coluna:

`{\rm texto}`	`{\textrm texto}`	texto
`{\bf texto}`	`{\textbf texto}`	**texto**
`{\sl texto}`	`{\textsl texto}`	*texto*
`{\sf texto}`	`{\textsf texto}`	texto
`{\it texto}`	`{\textit texto}`	*texto*
`{\tt texto}`	`{\texttt texto}`	`texto`

Exemplo 2.2.1 *Vejamos agora uma lição sobre formatação de textos. Tente produzir o código abaixo:*

Palavras **podem** *ser* escritas *em* VÁRIOS `formatos.`

```
(\rm Palavras} {\bf podem} {\sl ser} {\sf escritas}
      {\it em} {\sc vários} {\tt formatos.})
```

2.2.2 Tamanho da letra

Assim como o tipo de letra, também podemos alterar o tamanho da letra. Veja abaixo os comandos e seus respectivos resultados:

`{\tiny Texto}`	Texto
`{\scriptsize Texto}`	Texto
`{\footnotesize Texto}`	Texto
`{\small Texto}`	Texto
`{\normalsize Texto}`	Texto
`{\large Texto}`	Texto
`{\Large Texto}`	Texto
`{\LARGE Texto}`	Texto
`{\huge Texto}`	Texto
`{\Huge Texto}`	Texto

2.2.3 Cores

Algumas vezes queremos escrever textos coloridos em LaTeX, bom, para isso você precisará incluir os pacotes

- `\usepackage[dvips]{color}`
- `\usepackage{color}`

no preâmbulo e usar o comando `\color{cor}`.

Você pode usar as cores abaixo:

black	preto
red	vermelho
green	verde
blue	azul
cyan	azul claro
yellow	amarelo
magenta	rosa

Todo texto que estiver depois do comando **color** mudará de cor, para voltar a cor original use o comando novamente com a cor original do texto. Veja um exemplo abaixo:

Leia o texto abaixo:

Azul vermelho preto amarelo rosa verde azul claro.

O código segue abaixo:

```
\color{red}Azul \color{yellow}vermelho \color{green}preto
\color{blue}amarelo \color{cyan}rosa \color{magenta}verde
\color{black}azul claro.
```

2.3 Seções, capítulos, parágrafos

2.3.1 Seções

Um texto pode ser subdividido em seções, subseções e subsubseções. Fazemos isso com os comandos `\section{...}`, `\subsection{...}` e `\subsubsection{...}`, respectivamente.

Veja abaixo um exemplo:

```
\section{Textos em $\LaTeX$}
Texto 1
\subsection{Ambiente matemático}
Texto 2
\subsubsection{Letras gregas}
Texto 3
\section{Textos em Microsoft Word}
Texto 4
\subsection{Equation}
Texto5
\subsection{Tabelas}
Texto 6
```

Produz:

1 Textos em LaTeX

Texto 1

1.1 Ambiente matemático

Texto 2

Letras gregas

Texto 3

2 Textos em Microsoft Word

Texto 4

2.1 Equation

Texto 5

2.2 Tabelas

Texto 6

Como vimos o LaTeX enumera as seções e subseções deixando o seu texto mais organizado. Veremos adiante que podemos dividir um texto de outras formas.

2.3.2 Capítulos

Nas classes **book** e **report** podemos utilizar o comando `\chapter{...}`, para criar capítulos. Segue abaixo um exemplo:

```
\chapter{Criando textos}
Podemos criar textos interessantes.
\section{Textos em $\LaTeX$}
Texto 1
\subsection{Ambiente matemático}
Texto 2
```

Produz:

Capítulo 1

Criando textos

Podemos criar textos interessantes.

1.1 Textos em LaTeX

Texto 1

1.1.1 Ambiente matemático

Texto 2

2.3.3 Abstract

Geralmente em teses você encontra um texto chamado **Abstract**, que é um resumo da mesma. Então na classe **article** utilize o comando `\begin{abstract}...\end{abstract}`. Segue abaixo um exemplo:

```
\begin{abstract} H. J. Munkholm obtained a generalization
for topological manifolds of the famous Borsuk-Ulam type theorem
proved by Conner and Floyd. The purpose of this paper is to prove a
version of Conner and Floyd's theorem for generalized manifolds.
\end{abstract}
```

E produz:

Abstract. H. J. Munkholm obtained a generalization for topological manifolds of the famous Borsuk-Ulam type theorem proved by Conner and Floyd. The purpose of this paper is to prove a version of Conner and Floyd's theorem for generalized manifolds.

2.3.4 Parágrafos

Caso você queira criar um parágrafo o LaTeX possui o comando `\paragraph{...}`. Segue abaixo o Exemplo:

```
\paragraph{$\LaTeX$ no meio científico\\}
O LaTeX é uma forte ferramenta no meio científico ...
```

E produz:

LaTeX no meio científico
O LaTeX é uma forte ferramenta no meio científico ...

2.3.5 Partes

Podemos criar partes em nossos textos com o comando `\part{...}`. Segue abaixo o exemplo:

```
\part{Primeiro contato}
O início do uso do LaTeX
\part{Nem tudo são flores}
Chegamos na parte difícil do curso.
Estude mais agora
```

E produz:

Parte I
Primeiro contato

O início do uso do LaTeX

Parte II
Nem tudo são flores

Chegamos na parte difícil do curso. Estude mais agora.

2.4 Espaços

2.4.1 Espaços em branco

Espaços em branco consecutivos são considerados como se fossem um único espaço.

Exemplo 2.4.1 *O código abaixo:*

```
Texto    sem     espaços     entre    as     palavras.
```

Produz:

Texto sem espaços entre as palavras.

A maneira de acrescentar **espaços em branco** é colocando no texto uma ou várias barras invertidas, separadas entre si por espaços. Como no código abaixo:
O código abaixo:

```
Texto \ \ \ \ com \ \ \ \ \ espaços \ \ \ \ \ entre \ \ as \ \ \
palavras.
```

produz:

Texto   com   espaços   entre   as   palavras.

2.4.2 Nova linha

O final de linha pode ser sinalizado com duas barras invertidas consecutivas, ou seja `\\` , ou com um comando `\newline`. Você pode ”quebrar” uma linha com o comando `\linebreak`, e que difere dos comandos anteriores porque justifica a linha quebrada.

Veja o exemplo abaixo:

```
Vejamos o exemplo deste texto que \newline quebra uma linha.

Vejamos o exemplo deste texto que \linebreak quebra uma linha e a justifica.
```

Que produz:

Vejamos o exemplo deste texto que
quebra uma linha.

Vejamos o exemplo deste texto que
quebra uma linha e a justifica.

2.4.3 Nova página

Para quebrar uma página utilize o comando `\pagebreak`. Veja o exemplo abaixo:

```
Vamos quebrar a página, agora \pagebreak \\ Viu? A página foi quebrada.
```

Que produz:

Vamos quebrar a página, agora	Viu? A página foi quebrada.

2.5 Ambientes

A estrutura de um texto em linguagem TEX é baseada em pequenos ambientes, estes ambientes tem parâmetros de ajuste e funções bem definidas. A seguir é apresentado alguns dos principais ambientes divididos em categorias: alinhamento de texto (center, flushright, flushleft); listas (itemize, enumerate, description); e por fim o verbatim.

Em geral os ambientes são iniciados e terminados da seguinte maneira:

```
\begin{ambiente}[especificação]

Aqui você coloca o texto que recebera a formatação do ambiente.

\end{ambiente}[parâmetros]
```

2.5.1 Alinhamento

Os principais tipo de alinhamentos são: centralizado, a direita e a esquerda. Observe que é necessário terminar um ambiente iniciado.

```
\begin{flushleft}
Este texto está a esquerda.
\end{flushleft}
```

```
\begin{center}
Este texto está centralizado.
\end{center}
```

```
\begin{flushright}
Este texto está direita.
\end{flushright}
```

O resultado dessa formatação é mostrado a seguir:

Este texto está a esquerda.

Este texto está centralizado.

Este texto está a direita.

Observe que alguns ambientes não requerem parâmetros. Caso você omita parâmetros em ambientes que necessitem dos mesmos, o compilador LaTeX usará a configuração padrão.

2.5.2 Verbatim

O ambiente *verbatim* é um ambiente especial. Todos os ambientes que trabalhamos até agora determinavam uma certa formatação ao texto, ao contrário do que temos visto esse novo ambiente produz um texto sem qualquer tipo de formatação. O resultado impresso será exatamente o texto escrito dentro do ambiente, seja símbolos ou texto. O ambiente é criado com o seguinte comando:

Para a confecção dessa apostila esse ambiente foi amplamente utilizado. Cada exemplo de comando ou sintaxe TeX encontrado nessa apostila é escrito dentro de um ambiente verbatim. Para textos pequenos (uma linha por exemplo) existe uma abreviação.

Segue dois exemplos.

2.5.3 Listas: itemize, enumerate e description

Os ambientes *itemize, enumerate e description*, formatam os três tipos básicos de listas: itens, enumeradas e descritivas. A formatação difere apenas no caractér que inicia um novo item da lista, a lista do tipo *itemize* é a lista mais simples

```
\begin{verbatim} ... seu texto ... \end{verbatim}
```

```
\verb|... seu texto ...|
```

e tem um caracter padrão, a enumerada usa a numeração arábica e a lista descritiva pode ser formatada com qualquer texto ou símbolo.

```
\item  Um item qualquer de uma lista de item
\item Um item semelhante para uma lista enumerada
\item[0.]  Um item descritivo com um símbolo numérico
```

Observe que assim como os demais tipos de ambientes é possível sobrepor vários tipos de listas e criar listas listas, isto é, criar listas dentro de listas independente do tipo.

```
\textbf{Desculpas por não ter terminado a tarefa de matemática}

\begin{itemize}
\item Hoje é dia do professor.
\item Acidentalmente eu dividi por zero e então meu caderno pegou fogo.
\item Eu coloquei meu trabalho dentro da garrafa de Klein.
\item O cachorro comeu meu caderno.
\end{itemize}
```

Desculpas por não ter terminado a tarefa de matemática

- Hoje é dia do professor.
- Acidentalmente eu dividi por zero e então meu caderno pegou fogo.
- Eu coloquei meu trabalho dentro da garrafa de Klein.
- Um cachorro comeu meu caderno.

```
\textbf{Os Dez Mandamentos da Matemática}
\begin{enumerate}
\item Começarás contagens pelo zero.
\item Derivarás e Igualarás a zero.
\item Amarás ao Cálculo como a ti mesmo.
\item Não esquecerás a constante de integração.
\item Não levantarás falso teorema.
\item Honrarás épsilon e delta.
\item Não dividirás por zero.
\item Usarás letras gregas difíceis como variável.
\item Não esquecerás o jacobiano.
\item Não perderás a piada.
\end{enumerate}
```

```
\verb|$x = \sqrt{3} + \pi \ y=a^b$|
```

$x = \sqrt{3} + \pi \ y = a^b.$

```
\begin{verbatin}
$x = \sqrt{3} + \pi \ y=a^b$
\end{verbatin}
```

$x = \sqrt{3} + \pi \ y = a^b.$

Os Dez Mandamentos da Matemática

0. Começarás a contagem pelo zero.
1. Derivarás e Igualarás a zero.
2. Amarás ao Cálculo como a ti mesmo.
3. Não esquecerás a constante de integração.
4. Não levantarás falso teorema.
5. Honrarás épsilon e delta.
6. Não dividirás por zero.
7. Usarás letras gregas difíceis como variável.
8. Não esquecerás o jacobiano.
9. Não perderás a piada.

```
\textbf{Definições usadas em uma aula de matemática}

\begin{description}
\item[Claramente:] Eu não estou com paciência de escrever passo por passo.
\item[Trivial:] Se eu tiver que provar isso você deve estar na sala errada.
\item [Sem perda de generalidade:] Certamente eu não sei
todos os casos então faço o primeiro e deixo como exercício os restantes.
\item[Verifique:] Essa é a parte enfadonha da prova. Faça
como exercício.
\item[Uma dica:] Provavelmente o jeito mais complicado de provar.
\item[Uma prova simples:] Menos de um terço da folha mas
requer dois anos extra de estudo apenas para compreender a sintaxe.
\item [Uma prova elegante:] Não requer conhecimento prévio
algum e tem menos de dez linhas.
\item[Pelo teorema anterior:] Eu não me lembro como chegamos
aqui (verfique!). Não estou certo do que temos feito até
agora, mas se o começo estiver correto o resultado segue.
\item [Prova em duas linhas:] Eu vou deixar tudo como conclusão.
\item[Finalmente:] Apenas mais dez passos provando se e
somente se pela contrapositiva e chegaremos ao resultado final.
\item[Prova omitida:] Acreditem, é verdade.
\end{description}
```

Definições freqüentemente usadas em uma aula de matemática

Claramente: Eu não estou com paciência de escrever passo a passo.

Trivial: Se eu tiver que provar isso você deve estar na sala errada.

Sem perda de generalidade: Certamente eu não sei todos os casos então faço o primeiro e deixo como exercício os restantes.

Verifique: Essa é a parte enfadonha da prova. Faça como exercício.

Uma dica: Provavelmente o jeito mais complicado de provar.

Uma prova simples: Menos de um terço da folha mas requer dois anos extra de estudo apenas para compreender a sintaxe.

Uma prova elegante: Não requer conhecimento prévio algum e tem menos de dez linhas.

Pelo teorema anterior: Eu não me lembro como chegamos aqui (verifique !) Não estou certo do que temos feito até agora, mas se o começo estiver correto o resultado segue.

Prova em duas linhas: Eu vou deixar tudo como conclusão.

Finalmente: Apenas mais dez passos provando “se e somente se” pela contra positiva e chegaremos ao resultado final.

Prova omitida: Acreditem, é verdade.

2.6 Corpos flutuantes

Corpos flutuantes são figuras ou tabelas, posicionadas de maneira a otimizar a ocupação das páginas. Estes elementos não podem conter quebras de página, e são inseridos pelo LaTeX em uma fila de colocação. De acordo com alguns parâmetros especificados pelo usuário, os corpos flutuantes são posicionados no documento. Para evitar que os corpos flutuantes tornem-se uma grande fonte de frustração (o LaTeX nunca os posiciona onde você deseja), é importante entender um pouco do mecanismo interno de posicionamento de *floats*. Os corpos flutuantes são ambientes, e podem ser iniciados, por exemplo, da seguinte forma:

```
\begin{figure}[especificação]  ou
\begin{table}[especificação ]
```

Onde o *especificação* é um parâmetro que o usuário define. Na Tabela 2.1 são mostrados esses parâmetros:

Parâmetro	Permissão para posicionar o corpo flutuante ...
h	no lugar onde ocorreu no texto.
	Funciona para pequenas tabelas ou figuras.
t	no topo de uma página.
b	na parte inferior de uma página.
p	em uma página especial contendo somente corpos flutuantes.
!	sem considerar alguns parâmetros que poderiam
	impedir esse corpo de ser posicionado .

Tabela 2.1: Parâmetros de Posicionamento.

O LaTeX posiciona todos *floats* que encontra de acordo com o parâmetro de posicionamento fornecido. Se um corpo flutuante não puder ser colocado na página atual, ele é inserido em uma fila de figuras ou de tabelas. Quando uma nova página é iniciada, o primeiro *float* é tratado como se tivesse ocorrido neste ponto do texto: o LaTeX tenta outra vez posicioná-lo de acordo com seu parâmetro (exceto 'h', que não é mais possível). Qualquer novo *float* que ocorra é colocado em sua respectiva fila. O A ordem de ocorrência dos corpos flutuantes é mantida estritamente, por isso uma figura que não pode ser posicionada "atrasa" a colocação de todas as figuras seguintes.

Mesmo sendo possível dar ao LaTeX uma única possibilidade de posicionamento, isso causa problemas. Se o *float* não pode ser posicionado no lugar especificado, ele fica preso, bloqueando todos os corpos flutuantes seguintes. Em particular, nunca deve-se usar a opção `[h]`, que nas últimas versões do LaTeX é substituída automaticamente por `[ht]`. Sob algumas circunstâncias, pode-se usar o comando `\clearpage`, que ordena que todas as figuras e tabelas nas filas sejam posicionadas, e inicia uma nova página.

Dentro dos ambientes *table* e *figure*, podemos usar os seguinte comando para colocar um título:

```
\caption{texto}
```

Ainda, é possível criar listas dos corpos flutuantes, assim como um sumário da seguinte maneira:

```
\listoftables    \listoffigures
```

2.6.1 Figuras

Existem diversos pacotes para a linguagem LaTeX que facilitam o trabalho de inserir figuras de diversos tipos e formatá-las corretamente em seu texto. Aqui, trabalharemos com o ambiente *figure* para isso. Vamos trabalhar com formatos de figuras *eps e pdf* (Encapsulated PostScript - EPS; Portable Document Format - PDF), para isso será necessário ter instalado o pacote *graphicx* no preâmbulo do documento. Neste caso, também pode usado um parâmetro opcional com o nome do driver a ser usado na conversão de DVI para PS ou PDF.

```
\usepackage{graphicx}
\usepackage[dvips]{graphicx}
\usepackage[pdftex]{graphicx}
```

A sintaxe do comando `\includegraphicx` é dada por

```
\includegraphics[opções]{nome do arquivo.extenção}
```

As opções são: height (altura), width (largura), scale (escala), angle (ângulo). O uso desse comando não se restringe ao ambiente *figure* no entanto uma forma eficiente de usar o `\includegraphics` é usá-lo em um ambiente figure. Assim, ele pode ser referenciado mais facilmente, pode ser legendado ou "flutuar" para um local conveniente no texto.

```
\begin{figure}[posicionamento]
...(comandos relacionados com a figura)...
\caption{Legenda da figura}
\label{marca}
\end{figure}
```

Parâmetro	Permissão para posicionar o corpo flutuante ...
h	here (aqui)
t	top (topo de uma página)
b	bottom (ao fundo de uma página)
p	page (em uma página só com objetos flutuantes)
!	esse comando pode ser usado junto com os acima para priorizar inserir o texto onde ele aparecer

Tabela 2.2: Alguns parâmetros de posicionamento.

Até agora temos visto comando básicos para inserção de figuras em texto, veremos agora como inserir texto ao redor de figuras e para isso precisaremos do pacote *wrapfig*.

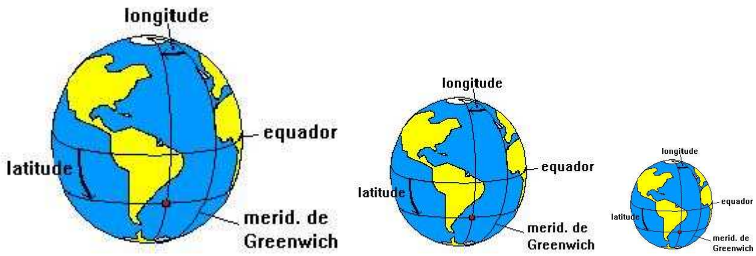


Figura 2.1: Um exemplo de figura com legenda e várias figuras com escala: 0.7, 0.5, 0.3.

```
\begin{figure}[ht]
  \centering
  \includegraphics[scale=0.7]{meridianos.eps}
  \includegraphics[scale=0.5]{meridianos.eps}
  \includegraphics[scale=0.3]{meridianos.eps}
\caption{Um exemplo de figura com legenda e várias figuras
com escala: $0.7$, $0.5$, $0.3$.}
\end{figure}
```

```
\begin{wrapfig}[nLinhas]{posicionamento}{largura}
...comandos...
\end{wrapfig}
```

Parâmetro	Permissão para posicionar o corpo flutuante ...
nLinhas	Parâmetro opcional que define o número de linhas do texto.
r	right (a direita)
l	left (a esquerda)
i	inside (dentro da página)
o	outside (fora da página)
largura	largura da figura

Tabela 2.3: Alguns parâmetros do ambiente *wrapfig*.

Definição: Meridiano (ou meridiano geográfico) é a linha imaginária que resulta de um corte efetuado num modelo geométrico da Terra por um plano que contém o seu centro. Quando esse modelo é uma esfera, o meridiano é uma semi-circunferência (180 graus); quando é um elipsóide de revolução, é uma semi-elipse. Em ambos os casos, o meridiano contém os pólos e é perpendicular a todos os paralelos, e ao equador.

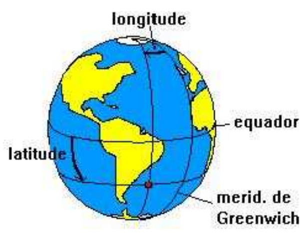


Figura 2.2: Figura ao lado de texto (*wrapfigure*)

2.6.2 Mini-pages

Para colocar duas ou mais figuras lado a lado, deve-se usar um único ambiente figure e colocar nele as figuras dentro de ambientes mini-page. Cada mini-page pode ter suas próprias legendas, rótulos, etc. A soma das larguras de todas as mini-pages não deve ultrapassar o tamanho da linha atual (`\linewidth`).

```
% Duas figuras lado a lado
\begin{figure}[!htb]
\begin{minipage}[b]{0.45\linewidth}
  includegraphics[width=\linewidth]{patu.eps}
  \caption{Figura da esquerda} \label{fig:patu}
  \end{minipage} \hfill
\begin{minipage}[b]{0.45\linewidth}
  \includegraphics[width=\linewidth]{catole.eps}
  \caption{Figura da direita} \label{fig:catole}
\end{minipage}
\end{figure}
```

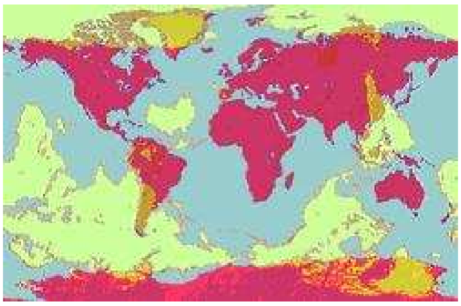


Figura 2.3: Figura da esquerda

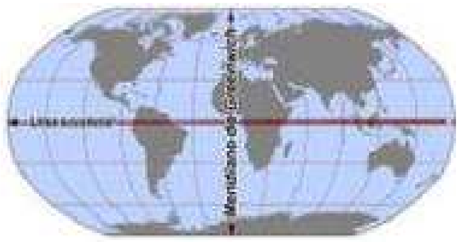


Figura 2.4: Figura da direita

Observe que em todos os casos, as figuras tinham o formato *.eps* ou *.pdf*. Para converter imagens de outros formatos para esses formatos utilize o prompt de comando do sistema operacional linux com o comando:

`convert arq.jpeg arq.eps` ou `convert arq.jpeg arq.pdf`

2.6.3 Tabelas

Para formatar textos em tabelas iremos utilizar o ambiente *tabular*, muito parecido com o ambiente *array* que será abordado adiante esse ambiente compõe o texto em colunas e linhas. Assim como outros ambientes anteriormente estudados existem parâmetros específicos que determinam qual o alinhamento das colunas bem como quando começa e termina uma linha ou coluna. Para terminar uma linha usa-se `\\` e para separar as colunas `&`. A seguir os parâmetros de alinhamento e número de colunas da tabela.

Para inserir bordas verticais na tabela adiciona-se `|` entre os parâmetros de alinhamento das colunas e `\hline` para bordas horizontais. A seguir a formatação da Tabela (2.4).

Parâmetro	Tipo de alinhamento
r	right (a direita)
l	left (a esquerda)
c	center (centralizado)

Tabela 2.4: Alguns parâmetros do ambiente *tabular* para tabelas.

```
\begin{tabular}{ll}
Parâmetro & Tipo de alinhamento \\
\hline
$r$ & right (a direita) \\
$l$ & left (a esquerda) \\
$c$ & center (centralizado) \\
\hline
\end{tabular}
```

Outra opção para a construção de tabelas é dada pelo ambiente `\begin{table}`, sendo que a tabela, neste caso, "flutua" no texto e o LaTeX escolhe a melhor localização para inseri-la. Esse outro ambiente possui legenda e parâmetro de posicionamento semelhante a uma figura. Segue um exemplo.

```
\begin{table}[!htb]
\centering
\begin{tabular}{l|r|c|r}
\hline
Esquerda & Direita & Centrado & Direita \\ \hline \hline
Um & Dois & Três & Quatro \\
i & ii & iii & iv \\
\hline
\end{tabular}
\caption{Usando o ambiente \emph{table}}
\label{tab:Tabe}
\end{table}
```

Esquerda	Direita	Centrado	Direita
Um	Dois	Três	Quatro
1	2	3	4
i	ii	iii	iv

Tabela 2.5: Usando o ambiente *table*

Tabelas mais sofisticadas mesclando linhas colunas ou parte delas podem ser feitas com os comandos `\cline` ou `\multicolumn`.

```
\multicolumn{num_colunas}{alinhamento}{texto}
\cline{coluna m a coluna n}
```

```
\begin{table}[!htb]
\centering
\begin{tabular}{c|l|l|c|c}
\hline
Nível & Ano & Curso & Inscritos & Selecionados \\
\hline
Mestrado & 2000 & Estatística & 15 & 10 \\
\cline{3-5} & & Genética & 20 & 10 \\
\cline{2-5} & 2001 & Estatística & 18 & 10 \\
\cline{3-5} & & Genética & 15 & 10 \\
\hline
\end{tabular} \caption{Seleção de pós-graduandos}
\end{table}
```

Nível	Ano	Curso	Inscritos	Selecionados
Mestrado	2000	Estatística	15	10
		Genética	20	10
	2001	Estatística	18	10
		Genética	15	10

Tabela 2.6: Seleção de pós-graduandos

```
\begin{table}[!htb]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
Atividades & \multicolumn{12}{|c|}{Meses} \\
\cline{2-13} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\hline 1 &   &   & X & X &   &   &   &   & X & X & X &   \\
\hline 2 &   &   & X & X & X & X & X & X & X & X & X &   \\
\hline 3 &   &   & X & X &   &   &   &   &   &   &   &   \\
\hline
\end{tabular} \caption{\emph{Cronograma de Atividades 2008}}
\end{table}
```

Atividades	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
1			X	X					X	X	X	
2			X	X	X	X	X	X	X	X	X	
3			X	X								

Tabela 2.7: *Cronograma de Atividades 2008*

2.7 Finalizando um texto.

2.7.1 Sumário.

Sumários e listas de figuras e tabelas são criados de forma muito semelhante no LaTeX. A maior parte do trabalho é feita pelo LaTeX, que se encarrega de

coletar as entradas e gerar o sumário/lista incluindo os números de página corretos. Porém, para isso é necessário rodar o LaTeX duas ou até três vezes para que uma modificação no sumário seja incluída e apareça com o número de página correto.

Os comandos abaixo são auto-explicativos. é tudo o que é necessário para incluir um sumário ou uma lista de figuras ou tabelas.

`\tableofcontents` `\listoffigures` `\listoftables`

Às vezes é necessário adicionar algo no sumário que o LaTeX não coloca por padrão. Em geral, isso acontece com capítulos ou seções geradas com `\chapter*` ou `\section*`, que não são incluídos no sumário por padrão. Para fazer isso, adicione esse comando após definir a seção:

`\addcontentsline{toc}{nível}{texto}`

Onde nível é o nível da entrada, e é chapter, section, subsection, conforme apropriado. texto é o texto que vai aparecer.

é possível também usar o mesmo comando usando lof ou lot ao invés de toc no primeiro argumento para incluir entradas na lista de figuras ou de tabelas, respectivamente. nível deverá ser então figure ou table respectivamente.

O comando acima gera, no arquivo que é posteriormente processado para gerar o índice, uma entrada usando o comando cujo funcionamento é descrito mais adiante. Se você quiser incluir qualquer outro texto ou comando, use:

`\addtocontents{toc}{texto}`

2.7.2 Configurando o Sumário

O sumário ou lista é gerado lendo-se um arquivo com a extensão .toc (para sumário), .lof (para lista de figuras) ou .lot (para lista de tabelas), que pode conter quaisquer comandos LaTeX válidos. Mas em geral ele consistirá de uma série de linhas contendo o comando `\contentsline`:

`\contentsline{nível}{texto}{página}`

Os argumentos são o nível da entrada (section, subsection, figure, etc. esse nível determina o comando que será chamado em seguida), o texto a ser impresso e o número da página a ser impressa.

O que o comando `\contentsline` faz é chamar um comando com o nome `\l@nível`, com dois argumentos: o texto e o número da página. Esses comandos, um para cada tipo, é que são responsáveis por formatar uma linha com uma entrada no sumário. Assim, caso se deseje mudar a maneira como as linhas das seções são mostradas, deve-se alterar o comando `\l@section`.

O argumento texto de posteriormente passado para `\l@nível`) em geral tem a forma

`\numberline{número}título`

Para mostrar o número da seção seguido do texto.

Um comando útil na definição dos comandos `\l@nível` é `\@dottedtocline`, que gera uma linha pontilhada com o número, texto e número da página:

```
\@dottedtocline{nível}{identação}{largura}{texto}{página}
```

Os argumentos são: nível é o nível **numérico** da seção, quanto maior, mais aninhada ela está. identação é o espaço deixado desde a margem esquerda. largura é a largura do box que contém o número da seção, caso o comando `\numberline` seja usado. Os dois últimos argumentos são os mesmos de `\l@nível`, o que facilita a definição dos comandos.

Eis alguns exemplos:

```
\renewcommand{\l@section}{\@dottedtocline{1}{1.5em}{2.3em}}
\renewcommand{\l@subsection}{\@dottedtocline{2}{2.8em}{3.2em}}
```

2.7.3 Índice remissivo.

O índice remissivo é aquele índice que fica no final do trabalho e remete aos tópicos desenvolvidos no trabalho na ordem em que eles aparecem. Um índice remissivo é construído usando-se o pacote makeidx, use o seguinte comando no preâmbulo do seu trabalho.

Comando	Lugar onde deve ser adicionado
`\usepackage{makeidx}`	No preâmbulo.
`\makeidx`	Também no preâmbulo.
`\printindex`	No local onde será impresso o índice.
`\index{entrada}`	Para cada palavra que você quiser colocar é preciso mapeá-la no texto dessa maneira.

Tabela 2.8: Como adicionar um índice remissivo.

A criação do índice remissivo segue alguns passos e cria alguns arquivos no diretório do arquivo. Entenda porque é necessário compilar seu código algumas vezes para que o índice aparece no lugar desejado.

Quando o pacote *makeid* é usado e o LaTeX é executado, ele cria um arquivo de extensão *.idx*. Ao ser criado o arquivo de extensão *.idx*, deve-se recompilar o código para que seja criado, a partir do arquivo *.idx*, um arquivo de extensão *.ind*. Uma vez criado o arquivo *.ind*, basta executar o LaTeX novamente para ter o índice remissivo criado no lugar escolhido.

2.7.4 Referências bibliográficas.

Aqui apresentaremos uma maneira muito eficiente e inteligente de criar a lista de referência de obras. Para tanto criaremos uma base de dados com as obras em um arquivo separado e somente as obras citadas e referenciadas no texto são inseridas no final do trabalho quando o comando de criação é executado.

Criando a base de referências no arquivo *bibliografia.bib*

As bibliografias em LaTeX são formatadas automaticamente, sendo apenas necessário introduzir os dados corretamente.

A verdade é que esses dados podem ser introduzidos de maneiras distintas, uma das melhores maneiras de o fazer é criando uma base de dados com todos os dados bibliográficos que vamos utilizar. Pode parecer a forma mais difícil de o fazer mas a verdade é que podemos criar uma base de dados que serve para vários documentos e portanto poupamos trabalho para o futuro.

é possível utilizar o editor de LaTeX para criar a nossa base de dados. Para isso fazemos o seguinte:

- Criamos um novo documento e gravamo-lo com o nome que quisermos mas com a extensão *.bib*. Exemplo: bibliografia.bib.
- Depois introduzimos os itens de bibliografia que queremos.
- Depois de escolhermos que tipo de documento queremos referir, completamos os espaços com as informação que tivermos. Se não quisermos preencher alguma dessa informação, apagamos toda a linha onde ela deveria constar.
- Depois de feita a inserção dos dados bibliográficos gravamos o ficheiro *.bib* e mantemo-lo aberto enquanto trabalhamos no documento *.tex* para que a citação seja mais rápida.

é possível editar o arquivo *.bib* em alguns editores LaTeX como o *WinEdit*.

Inserindo referências do arquivo *bibliografia.bib*

Depois de concebida a base de dados com todos os dados bibliográficos dos livros que queremos referir, vamos com certeza querer citá-los no documento. Para isso temos que seguir alguns procedimentos.

Devemos em primeiro lugar definir o estilo de bibliografia, ou seja, a formatação. Isso faz-se com a linha:

```
\bibliographystyle{estilo}
```

Os estilos disponíveis são imensos e podem encontrar alguns exemplos em *http://www.cs.stir.ac.uk/ kjt/software/latex/showbst.html*.

Depois, no local devido, devem inserir a bibliografia. Supondo que a base de dados se chama bibliografia.bib, introduzimos:

```
\bibliography{bibliografia}
```

A citação das obras em si faz-se usando o comando `\cite{nome}`, onde nome será a identificação dada na base de dados à obra em questão.

Capítulo 3

Ambiente Matemático

Estamos num ambiente matemático quando usamos cifrões `$...$`, `$$ ... $$` ou `\begin{equation}...\end{equation}`, neste ambiente as letras assumem o formato itálico. Toda fórmula deve ser digitada no modo matemático, veremos nos tópicos a seguir como podemos criar nossas fórmulas e expressões matemáticas.

3.1 Principais Elementos

Vejamos nesta seção os alguns dos principais elementos de um ambiente matemático.

3.1.1 Produzindo um Texto

Como já foi dito, para produzir um texto matemático, podemos utilizar três métodos.

Cifrões Simples

Cifrão simples é o método mais utilizado para fazer um texto matemático, o usamos quando queremos inserir um objeto matemático dentro de um texto comum.

```
Temos agora um texto comum com o polinômio $x^{2} - x + 1$.
```

E logo abaixo o texto formatado:

Temos agora um texto comum com o polinômio $x^2 - x + 1$.

Cifrões Duplos

Cifrão duplo é usado quando queremos inserir um objeto matemático em destaque fora de um texto comum. Mais a diante veremos que estes cifrões mudam alguns comandos.

```
Temos agora um texto comum com o polinômio $$x^{2} - x + 1$$.
```

E logo abaixo o texto formatado:

Temos agora um texto comum com polinômio

$$x^2 - x + 1.$$

Ambiente Equation

O Ambiente equation produz o mesmo que os cifrões duplos, numerando a equação ou fórmula.

```
Temos agora um texto comum com o polinômio
    \begin{equation}
        x^{2} - x + 1
    \end{equation}
```

E logo abaixo o texto formatado:

Temos agora um texto comum com o polinômio

$$x^2 - x + 1 \tag{3.1}$$

3.1.2 Letras gregas

Letras gregas são produzidas,em geral, colocando uma barra invertida antes do nome da respectiva letra, mas algumas delas são produzidas com um formato um pouco diferente. Se quisermos produzir uma letra minúscula, então devemos escrevê-la com letras minúsculas. Se quisermos que a letra seja maiúscula, devemos escrever *apenas* a primeira letra maiúscula e as demais minúsculas. Não podemos produzir todas as letras gregas em maiúsculo, por isso segue abaixo uma tabela exemplificando e mostrando as possíveis letras gregas que podem ser produzidas.

`\alpha`	α	`\xi`	ξ
`\beta`	β	`\o`	ø
`\gamma`	γ	`\pi`	π
`\delta`	δ	`\varpi`	ϖ
`\epsilon`	ϵ	`\rho`	ρ
`\varepsilon`	ε	`\varrho`	ϱ
`\zeta`	ζ	`\sigma`	σ
`\eta`	η	`\varsigma`	ς
`\theta`	θ	`\tau`	τ
`\vartheta`	ϑ	`\tau`	τ
`\iota`	ι	`\upsilon`	υ
`\kappa`	κ	`\phi`	ϕ
`\lambda`	λ	`\varphi`	φ
`\mu`	μ	`\chi`	χ
`\nu`	ν	`\omega`	ω

Minúsculas

\Gamma	Γ	\Lambda	Λ
\Sigma	Σ	\Psi	Ψ
\Delta	Δ	\Xi	Ξ
\Upsilon	Υ	\Omega	Ω
\Theta	Θ	\Pi	Π
\Phi	Φ		

Maiúsculas

3.2 Expoentes e Índices

Muitas vezes utilizamos variáveis com expoentes ou índices, por exemplo o polinômio $P(x) = x^2$ ou a variável x_2. Vejamos agora o código para escrevermos este tipo de texto.

`$X^{k}$`	X^k
`$X_{k}$`	X_k

3.3 Frações

Frações são facilmente construídas com o comando:

```
$\frac{numerador}{denominador}$
```

3.4 Raízes

Raízes quadradas são construídas com comando:

```
$\sqrt{radicando}$
```

Se quisermos construir uma raiz n-ésima utilizamos o comando:

```
$\sqrt[n]{radicando}$
```

3.5 Somatórios, produtórios, uniões e intersecções

Mostraremos como fazer somatórios, produtórios, uniões e intersecções juntos, pois suas sintaxes são muito parecidas. Vejamos abaixo os comandos

```
$\sum_{limite inferior}^{limite superior}$
$\prod_{limite inferior}^{limite superior}$
$\bigcup_{limite inferior}^{limite superior}$
$\bigcap_{limite inferior}^{limite superior}$
```

Exemplo 3.5.1 *Vejamos agora um exemplo mostrando a aplicação dos comandos.*

```
$\sum_{n=0}^{\infty}$

$\prod_{n=1}^{5}$

$\bigcup_{n=0}^{100}$

$\bigcap_{n=0}^{\infty}$
```

O código acima produz:

$\sum_{n=0}^{\infty}$

$\prod_{n=1}^{5}$

$\bigcup_{n=0}^{100}$

$\bigcap_{n=0}^{\infty}$

Se usarmos dois cifrões aumentaremos o tamanho dos símbolos, mas os deixaremos destacados numa linha.

Exemplo 3.5.2 *Texto comum com um somatório* `$$\sum_{i=0}^{\infty}a_{i}$$`
O código acima produz:

Texto comum com um somatório

$$\sum_{i=0}^{\infty} a_i$$

O mesmo efeito acontece se usarmos o ambiente equation, no entanto, podemos escrever de outra forma, preservando o tamanho do símbolo e o deixando na mesma linha de um texto comum.

3.5.1 displaystyle

Como dissemos o comando **display** altera o tamanho do símbolo, bem como, a posição dos indices sem deixá-lo sozinho numa linha. Faremos um exemplo, mostrando como isso acontece.

Exemplo 3.5.3 *Um somatório* $\displaystyle\sum_{n=1}^{\infty}$, *um produtório* $\displaystyle\prod_{n=1}^{5}$, *uma união* $\displaystyle\bigcup_{n=0}^{100}$ *e uma intersecção* $\displaystyle\bigcap_{n=0}^{\infty}$.

3.6 Limites, derivadas e integrais

Não se assuste! Não faremos aqui um curso de cálculo, apenas mostraremos como escrever os objetos mais importantes de um curso de cálculo.

3.6.1 Limites

De modo geral, os comandos no ambiente matemático não são muito diferentes, vejamos abaixo o comando para escrever um limite e observemos que não é muito diferente dos outros que já estudamos.De um modo geral, temos o código:

`$\lim_{variável\to valor}função$`

`$\lim_{x \to a}f(x)=f(a)$`	$\lim_{x\to a} f(x) = f(a)$
`$\displaystyle \lim_{x \to a}f(x)=f(a)$`	$\displaystyle\lim_{x\to a} f(x) = f(a)$

3.6.2 Derivadas

Veremos agora os comandos para podermos escrever derivadas. é bom lembrar que a utilizando o expoente `^{'}` podemos escrever as derivadas de uma função de uma variável, mas estamos interessados em outras notações.

```
$\frac{dy}{dx}

$\frac{\partial f}{\partial x}(a, b)$

$f^{'}(x)+f^{''}(x)+f^{'''}(x)$
```

$\frac{dy}{dx}$

$\frac{\partial f}{\partial x}(a, b)$

$f^{'}(x)+f^{''}(x)+f^{'''}(x)$

3.6.3 Integrais

O comandos para integrais são muito parecidos com os de somatório, então esta parte será praticamente um *"review"* tópico 3.5. Segue abaixo o comando geral para construir integrais.

```
$\int_{limite inferior}^{limite superior}$
```

```
$\int_{a}^{b}$ f(x)= F(a)-F(b)$

$\displaystyle \int_{a}^{b} f(x)= F(a)-F(b)$

$\displaystyle \int_{a}^{\infty} f(x)=
 \displaystyle \lim_{b \to \infty} \displaystyle \int_{a}^{b}$

$\displaystyle \oint_C u(x, y) dx + v(x, y)dy$

$\displaystyle \int \! \! \! \int_\Omega f(u)g(v) du dv$
```

$\int_{a}^{b} f(x)= F(a)-F(b)$

$\displaystyle \int_{a}^{b} f(x)= F(a)-F(b)$

$\displaystyle \int_{a}^{\infty} f(x)= \displaystyle \lim_{b \to \infty} \displaystyle \int_{a}^{b} f(x)$

$\displaystyle \oint_C u(x, y) dx + v(x, y)dy$

$\displaystyle \int \! \! \! \int_\Omega f(u)g(v) du dv$

Como vimos acima, quando usamos integrais múltiplas, devemos utilizar `\!` para diminuir os espaços entre as integrais.

3.7 Coeficientes binomiais

Vejamos agora como produzir coeficientes binomiais, como $\binom{n}{k}$. Segue abaixo o comando geral:

```
{n \choose k}
```

O exemplo abaixo mostra como podemos utilizar este comando

```
$${n+1\choose k}={n\choose k}+{n\choose k-1}$$
```

e produz

$${n+1\choose k}={n\choose k}+{n\choose k-1}$$

3.8 Pontos, espaços e texto no modo matemático

Geralmente em textos matemáticos utilizamos pontos, veremos como criar alguns deles. Segue abaixo os comandos:

`$\cdot$`	$\cdot$
`$\ldots$`	$\ldots$
`$\cdots$`	$\cdots$
`$\vdots$`	$\vdots$
`$\ddots$`	$\ddots$

Vejamos agora como criar espaços **verticais** e **horizontais** num texto. Temos basicamente dois comandos, `\vspace` e `\hspace`, segue abaixo a sintaxe geral:

```
$\vspace{espaço}$    $\hspace{espaço}$
```

Por exemplo `\vspace{1cm}` gera um espaço vertical de 1 cm, enquanto `hspace{0.5cm}` gera um espaço horizontal de 0.5 cm.

Quando começamos um parágrafo, não conseguimos utilizar os comando acima, então podemos forçar um espaçamento incluindo * depois dos comandos. Por exemplo: `\hspace*{3cm}`

Podemos escrever textos em ambientes matemáticos, para isso use o comando `\mbox{texto}`.

Veja o exemplo abaixo:

```
$$
x_{1}=\frac{-b-\sqrt{b^{2}-4ac}}{2a} \quad\mbox{e}\quad
x_{2}=\frac{-b+\sqrt{b^{2}-4ac}}{2a}.
$$
```

$$x_{1}=\frac{-b-\sqrt{b^{2}-4ac}}{2a} \quad\mbox{e}\quad x_{2}=\frac{-b+\sqrt{b^{2}-4ac}}{2a}.$$

3.9 Símbolos matemáticos

Exibiremos agora algumas tabelas com os principais símbolos matemáticos. o leitor não precisa se preocupar em memorizar estes símbolos, pois os editores de LaTeX os contém.

Símbolos de Operações Binárias

`$\pm$`	$\pm$	`$\cap$`	$\cap$	`$\diamond$`	$\diamond$	`$\oplus$`	$\oplus$
`$\mp$`	$\mp$	`$\cup$`	$\cup$	`$\bigtriangleup$`	$\bigtriangleup$	`$\ominus$`	$\ominus$
`$\times$`	$\times$	`$\uplus$`	$\uplus$	`$\bigtriangledown$`	$\bigtriangledown$	`$\otimes$`	$\otimes$
`$\div$`	$\div$	`$\sqcap$`	$\sqcap$	`$\triangleleft$`	$\triangleleft$	`$\oslash$`	$\oslash$
`$\ast$`	$\ast$	`$\sqcup$`	$\sqcup$	`$\triangleright$`	$\triangleright$	`$\odot$`	$\odot$
`$\star$`	$\star$	`$\vee$`	$\vee$	`$\lhd$`	$\lhd$	`$\bigcirc$`	$\bigcirc$
`$\circ$`	$\circ$	`$\wedge$`	$\wedge$	`$\rhd$`	$\rhd$	`$\dagger$`	$\dagger$
`$\bullet$`	$\bullet$	`$\setminus$`	$\setminus$	`$\unlhd$`	$\unlhd$	`$\ddagger$`	$\ddagger$
`$\cdot$`	$\cdot$	`$\wr$`	$\wr$	`$\unrhd$`	$\unrhd$	`$\amalg$`	$\amalg$

Símbolos de Relações

`$\leq$`	$\leq$	`$\geq$`	$\geq$	`$\equiv$`	$\equiv$	`$\models$`	$\models$
`$\prec$`	$\prec$	`$\succ$`	$\succ$	`$\sim$`	$\sim$	`$\perp$`	$\perp$
`$\preceq$`	$\preceq$	`$\succeq$`	$\succeq$	`$\simeq$`	$\simeq$	`$\mid$`	$\mid$
`$\ll$`	$\ll$	`$\gg$`	$\gg$	`$\asymp$`	$\asymp$	`$\parallel$`	$\parallel$
`$\subset$`	$\subset$	`$\supset$`	$\supset$	`$\approx$`	$\approx$	`$\bowtie$`	$\bowtie$
`$\subseteq$`	$\subseteq$	`$\supseteq$`	$\supseteq$	`$\cong$`	$\cong$	`$\Join$`	$\Join$
`$\sqsubset$`	$\sqsubset$	`$\sqsupset$`	$\sqsupset$	`$\neq$`	$\neq$	`$\smile$`	$\smile$
`$\sqsubseteq$`	$\sqsubseteq$	`$\sqsupseteq$`	$\sqsupseteq$	`$\doteq$`	$\doteq$	`$\frown$`	$\frown$
`$\in$`	$\in$	`$\ni$`	$\ni$	`$\propto$`	$\propto$	=	$=$
`$\vdash$`	$\vdash$	`$\dashv$`	$\dashv$	<	$<$	>	$>$

Flechas

`$\leftarrow$`	$\leftarrow$	`$\longleftarrow$`	$\longleftarrow$	`$\uparrow$`	$\uparrow$
`$\Leftarrow$`	$\Leftarrow$	`$\Longleftarrow$`	$\Longleftarrow$	`$\Uparrow$`	$\Uparrow$
`$\rightarrow$`	$\rightarrow$	`$\longrightarrow$`	$\longrightarrow$	`$\downarrow$`	$\downarrow$
`$\Rightarrow$`	$\Rightarrow$	`$\Longrightarrow$`	$\Longrightarrow$	`$\Downarrow$`	$\Downarrow$
`$\leftrightarrow$`	$\leftrightarrow$	`$\longleftrightarrow$`	$\longleftrightarrow$	`$\updownarrow$`	$\updownarrow$
`$\Leftrightarrow$`	$\Leftrightarrow$	`$\Longleftrightarrow$`	$\Longleftrightarrow$	`$\Updownarrow$`	$\Updownarrow$
`$\mapsto$`	$\mapsto$	`$\longmapsto$`	$\longmapsto$	`$\nearrow$`	$\nearrow$
`$\hookleftarrow$`	$\hookleftarrow$	`$\hookrightarrow$`	$\hookrightarrow$	`$\searrow$`	$\searrow$
`$\leftharpoonup$`	$\leftharpoonup$	`$\rightharpoonup$`	$\rightharpoonup$	`$\swarrow$`	$\swarrow$
`$\leftharpoondown$`	$\leftharpoondown$	`$\rightharpoondown$`	$\rightharpoondown$	`$\nwarrow$`	$\nwarrow$
`$\rightleftharpoons$`	$\rightleftharpoons$	`$\leadsto$`	$\leadsto$		

Miscelânea de Símbolos

`$\ldots$`	$\ldots$	`$\cdots$`	$\cdots$	`$\vdots$`	$\vdots$	`$\ddots$`	$\ddots$
`$\aleph$`	$\aleph$	`$\prime$`	$\prime$	`$\forall$`	$\forall$	`$\infty$`	∞
`$\hbar$`	$\hbar$	`$\emptyset$`	$\emptyset$	`$\exists$`	$\exists$	`$\Box$`	$\Box$
`$\imath$`	$\imath$	`$\nabla$`	∇	`$\neg$`	$\neg$	`$\Diamond$`	$\Diamond$
`$\jmath$`	$\jmath$	`$\surd$`	$\surd$	`$\flat$`	$\flat$	`$\triangle$`	$\triangle$
`$\ell$`	ℓ	`$\top$`	$\top$	`$\natural$`	$\natural$	`$\clubsuit$`	$\clubsuit$
`$\wp$`	$\wp$	`$\bot$`	$\bot$	`$\sharp$`	$\sharp$	`$\diamondsuit$`	$\diamondsuit$
`$\Re$`	$\Re$	`$\|$`	$\|$	`$\backslash$`	$\backslash$	`$\heartsuit$`	$\heartsuit$
`$\Im$`	$\Im$	`$\angle$`	$\angle$	`$\partial$`	∂	`$\spadesuit$`	$\spadesuit$
`$\mho$`	$\mho$	.	.	+	+	-	-

3.10 Mudança dos Tipos de Letras no Modo Matemático

Diversos tipos de letras maiúsculas podem ser usados no modo matemático. Os mais usados podem ser usados tipos caligráficos ou tipos para denotar conjuntos numéricos. Importante notar que alguns tipos requerem a instalação de pacotes tais como *amssymb*.

`\cal{A}`	$\mathcal{A}$
`\mathcal{B}`	$\mathcal{B}$
`\mathbb{N}`	$\mathbb{N}$

3.11 Funções Matemáticas

O LaTeX possui vários tipos de funções pré-definidas como log, exp, cos. Se o nome da função não estiver pré-definido, pode-se defini-lo facilmente com o comando `\textrm{nome dafunção}`. A seguir alguns exemplos de funções do LaTeX.

`\log`	log	`\exp`	exp
`\cos`	cos	`\arccos`	arccos
`\sin`	sin	`\arcsin`	arcsin
`\tan`	sin	`\arctan`	arctan
`\cot`	cot	`\textrm{arccot}`	arccot
`\sec`	sec	`\textrm{arcsec}`	arcsec
`\csc`	csc	`\textrm{arccsc}`	arccsc
`\cosh`	cosh	`\textrm{arccosh}`	arccosh
`\sinh`	sinh	`\textrm{arcsinh}`	arcsinh
`\tanh`	tanh	`\textrm{arctanh}`	arctanh
`\max`	max	`\min`	min
`\lg`	lg	`\ln`	ln
`\sup`	sup	`\inf`	inf
`\dim`	dim	`\det`	det

3.12 Ambiente *Equation*

Em um texto matemático é comum criar equações e fórmulas, por vezes também é necessário citar, comentar ou referenciar estas fórmulas e equações anteriormente definidas. Em LaTeX o ambiente *equation* é uma estrutura para definir essas equações e fórmulas. A numeração é automática e a equação pode ser facilmente referenciada no texto com o mesmo comando de referências anteriormente estudado. Para fazer essas referências utiliza-se o comando *label* assim como em figuras ou tabelas. O *Equation* é muito mais do que um simples *Displaymath*.

```
\begin{equation}
a^2 + b^2 = c^2
\end{equation} \label{eq:pitagoras}
```

$$a^2 + b^2 = c^2 \tag{3.2}$$

O ambiente *eqnarray* pode ser usado para numerar fórmulas longas, que se "espalha" por mais de uma linha. Sua sintaxe é parecida com a do ambiente *array*. Cada final de linha recebe uma numeração, exceto aquelas assinaladas com um comando `\nonumber`.

```
\begin{eqnarray}
x & = & a +b+c+d+\nonumber \\
& & e + f + g + h + i \\
& > & y + z + 1
\end{eqnarray}
```

$$\begin{aligned} x &= a + b + c + d + \\ & \quad e + f + g + h + i && (3.3) \\ &> y + z + 1 && (3.4) \end{aligned}$$

3.13 Parênteses, colchetes e chaves em fórmulas

Alguns delimitadores podem ser usados em vários tamanhos, ajustando-se automaticamente ao tamanho da fórmula. Os delimitadores mais usados são os parênteses os colchetes e as chaves.

```
\left( ... \right)
\left[ ... \right]
\left{ ... \right}
```

Outra maneira de utilizar esses delimitadores é dada abaixo. No entanto o tamanho dos delimitadores é controlado manualmente.

```
\big( ... \big)
\bigg( ... \bigg)
\Big( ... \Big)
```

3.14 Vetores

Os vetores podem ser construídos com o comando `\vec` seguido da letra ou com um comando do tipo `\overrightarrow{}`, para gerar uma flecha maior sobre a expressão vetorial. Para colocar apenas uma barra superior sobre uma letra ou uma expressão use o comando `\bar` seguido da letra ou `\overline{expressão}`.

`\vec{v}`	$\vec{v}$
`\vec{w}`	$\vec{w}$
`\overrightarrow{2a+b-c}`	$\overrightarrow{2a+b-c}$
`\bar z = a - bi`	$\bar z = a - bi$
`\overline{z+2w}`	$\overline{z+2w}$

3.15 Matrizes

As matrizes são escritas em LaTeX dentro do ambiente *array*. A matriz é uma tabela de dados na qual podemos inserir símbolos matemáticos sem nos preocupar em usar o símbolo `$` único fato que o difere do ambiente *tabular* anteriormente estudado.

Para cada coluna da matriz podemos definir o alinhamento dos dados com uma letra: *c* para centralizar, *l* para alinhar a esquerda e *r* para alinhar a direita. Para separar as colunas usamos o símbolo `&` e para finalizar uma linha usamos o símbolo `\\`. De maneira análoga ao ambiente *tabular*. Dentro desse ambiente não é permitido "pular linha" e assim como no restante do código LaTeX valem as regras de espaço horizontal. Uma matriz é escrita da seguinte maneira:

```
\left( \begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{array}\right)
```

$$\left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \end{array}\right)$$

```
\left| \begin{array}{cc}
\displaystyle\frac{\partial u}{\partial x} &
\displaystyle\frac{\partial u}{\partial y}\\
\displaystyle\frac{\partial v}{\partial x} &
\displaystyle\frac{\partial v}{\partial y}\\
\end{array} \right|
```

$$\left| \begin{array}{cc} \displaystyle\frac{\partial u}{\partial x} & \displaystyle\frac{\partial u}{\partial y}\\ \displaystyle\frac{\partial v}{\partial x} & \displaystyle\frac{\partial v}{\partial y}\\ \end{array} \right|$$

A generalização desses conceitos vem com a prática, observe que não existe uma única maneira de formatar textos no entanto existe uma forma melhor para

cada ocasião. Por isso não se assuste com o tamanho ou a complexidade dos códigos.

Esse ambiente matricial também é usado para definir funções por partes.

```
|x| = \left\{ \begin{array}{ll}
+1 & \mbox{ se } x \geq 0; \\
-1 & \mbox{ se } x < 0.
\end{array} \right.
```

$$|x| = \left\{ \begin{array}{ll} +1 & \mbox{ se } x \geq 0; \\ -1 & \mbox{ se } x < 0. \end{array} \right.$$

3.16 Diagramas XY-pic

Para a construção de diagramas e grafos usaremos o pacote de comandos **XY-pic**. Faremos aqui uma introdução ao uso deste recurso. Para usá-lo, devemos colocar no preâmbulo do documento o comando

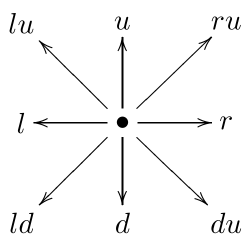
`\usepackage[all]{xy}`.

3.16.1 O comando xymatrix

O único comando do **XY-pic** que vamos explorar é o `\xymatrix`. Sua sintaxe é parecida com a do `\array` do LaTeX:

```
\xymatrix{
        a_11  &  a_12  &  ...  &  a_1n\\
        a_21  &  a_22  &  ...  &  a_2n\\
         ⋮        ⋮        ⋱        ⋮
        a_m1  &  a_m2  &  ...  &  a_mn}
```

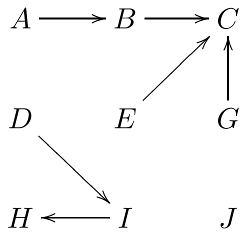
As entradas do `\xymatrix` podem ser vazias ou seguidas de . No caso da construção de diagramas, as são molduras ou setas nos mais diversos estilos, direções e sentidos. Uma seta é construída com um comando `\ar`, cuja sintaxe em sua forma mais simples é: `\ar[direção]` onde a direção é definida pelas letras d (para baixo), u (para cima), l (para a esquerda), r (para a direita) ou por combinações. Vejamos abaixo exemplos de combinações destas direções:



Vejamos agora um exemplo:

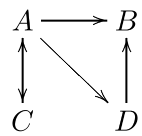
```
\xymatrix{
 A \ar[r]    &  B \ar[r]   &      C \\
 D \ar[rd]   &  E \ar[ru]  &  G \ar[u]\\
    H        &  I \ar[l]   &      J }
```

Que produz:



Podem partir várias setas de um mesmo elemento ou flechas com dois sentidos, ou seja, que partem de A para B e B para A. A única restrição ao uso das setas, é que elas devem apontar para uma entrada que realmente exista na matriz. Vejamos mais um exemplo do uso de flechas:

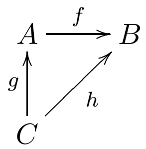
```
\xymatrix{
A \ar[r]\ar[d]  &     B \\
  C \ar[u]      &  D \ar[u] }
```



3.16.2 Índices nas setas

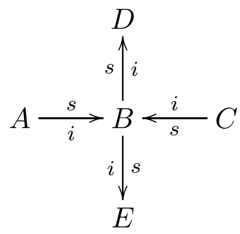
Na **Seção 3.2 Expoentes e índices**, aprendemos como escrever índices, a nomenclatura é a mesma, utilizamos `^{}` para *índices superiores* e `_{}` para *índices inferiores*, junto ao comando `\ar`. Vejamos um exemplo abaixo:

```
           \xymatrix{
        A \ar[r]^{f}          &  B \\
C \ar[u]^{g} \ar[u]_{ru}      &   }
```



Exemplo 3.16.1 *Os conceitos de "índice superior" e "índice inferior" dependem dos sentidos das setas. Neste exemplo, todos os "s" são índices superiores, enquanto que os "i" são inferiores.*

```
  \xymatrix{
       &            D \\
A \ar[r]^{s}_{i}       &  B \ar[d]^{s}_{i}\ar[u]^{s}_{i}
                       &          C \ar[l]^{s}_{i}             \\
       &            E }
```



Ainda podemos utilizar, no lugar de " `^` *" e "* `_` *" o parâmetro "* `|` *", então o índice é produzido dentro da seta. Por exemplo,* `\ar[r]|f` *mostra um f sobre uma seta como em* $U \;{-}f{\rightarrow}\; V$ *.*

Quando as entradas no comando `\xymatrix` *são muito extensas o índice não fica centralizado, para isso utilize o parâmetro "-" após "^" ou "* `_` *".*

Vejamos agora exemplos de aplicações dos parâmetros acima.

`A \times B \times C \ar[r]|f & D`	$A \times B \times C\; f\!\longrightarrow D$
`A \times B \times C \ar[r]|-f & D`	$A \times B \times C \;{-}f{\rightarrow}\; D$
`A \times B \times C \ar[r]^{f \circ g} & D`	$A \times B \times C^{f \circ g}\!\longrightarrow D$
`A \times B \times C \ar[r]^-{f \circ g} & D`	$A \times B \times C \xrightarrow{f \circ g} D$

3.16.3 Estilos das setas

Podemos mudar o estilo de uma seta com o uso do comando `@variante{cauda haste cabeça}`*, que é adicionado ao comando* `\ar`*. Vejamos abaixo os parâmetros que podem ser utilizados.*

- ***variante*** *é opcional e pode ser um dos caracteres* `_` *,* `^`*, 2 ou 3.*
- ***cauda*** *ou* ***cabeça*** *podem ser um dos caracteres* ><*ox+/()[]— ou espaço em branco.*
- ***haste*** *pode ser um dos caracteres* `~.:-=`

Veja a tabela abaixo com alguns exemplos:

<table>
<tr><th>estilo</th><th><code>A \ar estilo [r] B</code></th><th>estilo</th><th><code>A \ar estilo [r] B</code></th></tr>
<tr><td><code>@{->}</code></td><td>$\xymatrix{A \ar@{->}[r] & B}$</td><td><code>@{-->}</code></td><td>$\xymatrix{A \ar@{-->}[r] & B}$</td></tr>
<tr><td><code>@{=>}</code></td><td>$\xymatrix{A \ar@{=>}[r] & B}$</td><td><code>@{==>}</code></td><td>$\xymatrix{A \ar@{==>}[r] & B}$</td></tr>
<tr><td><code>@{.>}</code></td><td>$\xymatrix{A \ar@{.>}[r] & B}$</td><td><code>@{:>}</code></td><td>$\xymatrix{A \ar@{:>}[r] & B}$</td></tr>
<tr><td><code>@{~>}</code></td><td>$\xymatrix{A \ar@{~>}[r] & B}$</td><td><code>@{~~>}</code></td><td>$\xymatrix{A \ar@{~~>}[r] & B}$</td></tr>
<tr><td><code>@{-}</code></td><td>$\xymatrix{A \ar@{-}[r] & B}$</td><td><code>@{^{(}-o}</code></td><td>$\xymatrix{A \ar@{^{(}-o}[r] & B}$</td></tr>
<tr><td><code>@{.o}</code></td><td>$\xymatrix{A \ar@{.o}[r] & B}$</td><td><code>@{||.)}</code></td><td>$\xymatrix{A \ar@{||.)}[r] & B}$</td></tr>
<tr><td><code>@{=}</code></td><td>$\xymatrix{A \ar@{=}[r] & B}$</td><td><code>@{|-x}</code></td><td>$\xymatrix{A \ar@{|-x}[r] & B}$</td></tr>
<tr><td><code>@{|->>}</code></td><td>$\xymatrix{A \ar@{|->>}[r] & B}$</td><td><code>@{|.+}</code></td><td>$\xymatrix{A \ar@{|.+}[r] & B}$</td></tr>
<tr><td><code>@{(=)}</code></td><td>$\xymatrix{A \ar@{(=)}[r] & B}$</td><td><code>@{|->|}</code></td><td>$\xymatrix{A \ar@{|->|}[r] & B}$</td></tr>
<tr><td><code>@{^{(}->}</code></td><td>$\xymatrix{A \ar@{^{(}->}[r] & B}$</td><td><code>@{{*}-{*}}</code></td><td>$\xymatrix{A \ar@{{*}-{*}}[r] & B}$</td></tr>
<tr><td><code>@2{~>}</code></td><td>$\xymatrix{A \ar@2{~>}[r] & B}$</td><td><code>@3{<->}</code></td><td>$\xymatrix{A \ar@3{<->}[r] & B}$</td></tr>
<tr><td><code>@3{-}</code></td><td>$\xymatrix{A \ar@3{-}[r] & B}$</td><td><code>@3{.>}</code></td><td>$\xymatrix{A \ar@3{.>}[r] & B}$</td></tr>
<tr><td><code>@_{->}</code></td><td>$\xymatrix{A \ar@_{->}[r] & B}$</td><td><code>@^{->}</code></td><td>$\xymatrix{A \ar@^{->}[r] & B}$</td></tr>
<tr><td><code>@_{<->}</code></td><td>$\xymatrix{A \ar@_{<->}[r] & B}$</td><td><code>@^{<->}</code></td><td>$\xymatrix{A \ar@^{<->}[r] & B}$</td></tr>
<tr><td><code>@/_/</code></td><td>$\xymatrix{A \ar@/_/[r] & B}$</td><td><code>@/^/</code></td><td>$\xymatrix{A \ar@/^/[r] & B}$</td></tr>
</table>

Referências Bibliográficas

[1] Andrade, L. *Uma breve introdução ao LaTeX*. 2000.

[2] Costa, S. *Curso de introdução ao LaTeX*. 2002.

[3] Network, C. C. T. A. *The Not So Short Introduction to LaTeX*. 2003.

[4] Network, C. C. T. A. *TeX user groups around the world*. 2008.

[5] Neves, A. *O que eu vou aprendendo em LaTeX*. 2008.

[6] Santos, R. *Introdução ao LaTeX*. 2003.